



Universidade Federal de Alagoas (UFAL)
Campus Arapiraca



Programação Orientada a Objetos (POO)

02 - Programação estruturada com Java

Alexandre de Andrade Barbosa

`alexandre.barbosa@arapiraca.ufal.br`

Objetivos

Objetivos

- Descrever como criar e executar programas Java
- Apresentar os conceitos básicos de programação na linguagem Java:
 - tipos de dados e declaração de variáveis;
 - operadores relacionais, lógicos e matemáticos;
 - conversões entre tipos de dados;
 - entrada e saída de dados.

Programação estruturada com Java

Helo World!

Python

```
1 print 'Hello world!'
```

C

```
1 main() {  
2     printf(" Hello World!");  
3 }
```

Java

```
1 public class Executavel {  
2     public static void main(String[] args) {  
3         System.out.println(" Hello World!");  
4     }  
5 }
```

Programação estruturada com Java

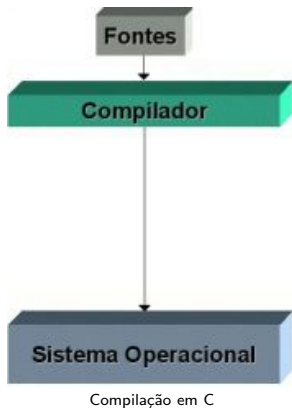
Compilação e Interpretação



Interpretação em Python

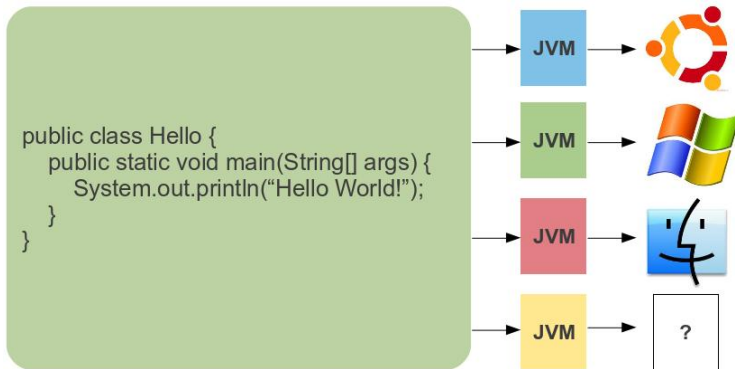
Programação estruturada com Java

Compilação e Interpretação



Programação estruturada com Java

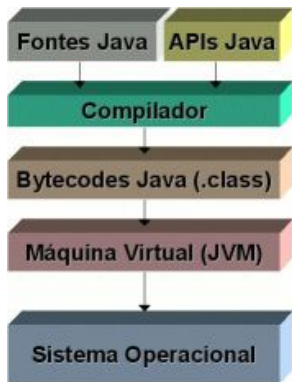
Programação e Java



Compilação e interpretação em Java

Programação estruturada com Java

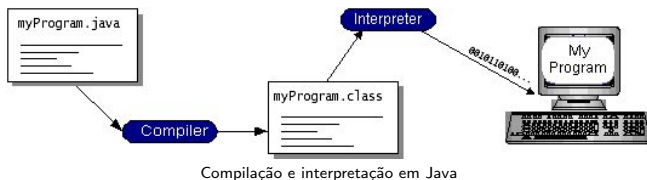
Programação e Java



Compilação e interpretação em Java

Programação estruturada com Java

Programação e Java



Programação estruturada com Java

Python e Java

Python	Java
Multiparadigma	Orientada a objetos
Tipagem forte e dinâmica	Tipagem forte e estática
Multiplataforma	Multiplataforma
Puramente interpretada	Híbrida (compilada e interpretada)
Ambiente interativo	-

Programação estruturada com Java

Programação e Java - Primeiro programa

```
1 class Primeiro {  
2     public static void main(String[] args) {  
3         System.out.println("Meu primeiro programa em  
4             Java!");  
5     }  
}
```

Programação estruturada com Java

Programação e Java - Primeiro programa

Compilando o programa

```
1 $ javac Primeiro.java
```

Executando / interpretando o programa

```
1 $ java Primeiro
```

Programação estruturada com Java

Programação e Java - Primeiro programa

Visão no terminal

```
1 $ javac Primeiro.java
2 $ java Primeiro
3 Meu primeiro programa em Java!
```

Programação estruturada com Java

Programação e Java - Primeiro programa

Observações:

- ❶ O nome do arquivo deve ser igual ao nome da classe
"Primeiro.java"
- ❷ "Função" (Método) main determina o início do programa
- ❸ Os blocos de código são delimitados por { e }
- ❹ Identação não determina os blocos (mas continue usando)
- ❺ Instruções são finalizadas com ;
- ❻ Sempre que alterar o código é necessário compilar novamente (javac)

Programação estruturada com Java

Tipos de Dados

- Python é uma linguagem com tipagem forte e dinâmica
- Java é uma linguagem com tipagem forte e estática
- C é uma linguagem com tipagem forte e estática
- É necessário declarar uma variável em Java e estabelecer seu tipo, que será imutável durante toda a execução do programa

Programação estruturada com Java

Tipos de Dados

- **int** - para representar números inteiros;
- **byte** - para representar números inteiros entre -128 e 127 ;
- **short** - para representar números inteiros entre -32.768 e 32.767 ;
- **long** - para representar números inteiros entre -2^{63} e $2^{63} - 1$

Programação estruturada com Java

Tipos de Dados

- **float** - para representar números reais (ponto-flutuante);
- **double** - para representar números reais (ponto-flutuante), com maior precisão que *float*;
- **boolean** - para representar valores lógicos: *true* (verdadeiro) ou *false* (falso);
- **char** - para representar um caractere Unicode;
- **String** - para representar cadeias de caracteres Unicode;

Programação estruturada com Java

Variáveis

Algumas **regras** para nomear variáveis em Java

- não podem iniciar com números
- caixa (alta ou baixa) diferencia os nomes (case sensitive)

Algumas **convenções** para nomear variáveis em Java

- **CamelCase** - nome deve iniciar com minúsculas e cada nome composto deve ter sua inicial maiúscula (ex. nome, nomeCompleto, nomePessoaFisica)
- não iniciar nomes de variáveis com `_` ou `$`

Observe todas as recomendações em:

<http://www.oracle.com/technetwork/java/codeconv-138413.html>

Programação estruturada com Java

Variáveis

```
1  public static void main(String[] args) {  
2      // define um inteiro  
3      int inteiro = 5;  
4      // define um real (float)  
5      float f = 5.0f; // observe o L no final  
6      // define um long  
7      long longo = 2550L; // observe o L no final  
8  
9      System.out.println(inteiro);  
10     System.out.println(f);  
11     System.out.println(longo);  
12 }
```

Programação estruturada com Java

Operadores

Atribuição (=)

```
1 int a = 5;
```

Comparação (==)

```
1 int a = 5;  
2 int b = 5;  
3 a == b resulta em true
```

Programação estruturada com Java

Operadores

Relação maior que (>)

```
1 5 > 2 resulta em true  
2 1 > 9 resulta em false;
```

Relação maior ou igual (>=)

```
1 5 >= 2 resulta em true  
2 1 >= 1 resulta em true
```

Relação menor que (<)

```
1 6 < 4 resulta em false  
2 ?1 < 0 resulta em true
```

Relação menor ou igual (<=)

```
1 2 <= 2 resulta em true  
2 1 <= 5 resulta em true
```

Programação estruturada com Java

Operadores

Aritmético adição (+)

```
1 6 + 3 resulta em 9
```

Aritmético multiplicação (*)

```
1 6 * 3 resulta em 18
```

Aritmético subtração (-)

```
1 6 - 3 resulta em 3
```

Aritmético divisão (/)

```
1 6 / 3 resulta em 2
```

Resto da divisão (%)

```
1 6 / 3 resulta em 0
```

Programação estruturada com Java

Operadores

Atribuição automática (+=, -=, *=, /=, %=)

```
1 int a = 6;  
2 int b = 3;  
3 b += a; // equivalente a b = b + a;
```

```
1 int a = 6;  
2 int b = 3;  
3 b /= a; // equivalente a b = b / a;
```

Programação estruturada com Java

Operadores

Incrementos ($x++$, $++x$)

```
1 // x++ Primeiro utiliza o valor da variável, depois
  incrementa
2 int x = 0;
3 System.out.println(x++);
4 System.out.println(x);
5 // ++x Primeiro incrementa o valor da variável, depois
  utiliza
6 int x = 0;
7 System.out.println(++x);
8 System.out.println(x);
```

Decrementos ($x--$, $--x$)

```
1 // x-- Primeiro utiliza o valor da variável, depois
  decrementa
2 int x = 2;
```

Programação estruturada com Java

Operadores

Operação lógica E (&&)

```
1 true && false resulta em false
2 true && true resulta em true
```

Operação lógica OU (||)

```
1 true || false resulta em true
2 true || true resulta em true
```

Operação lógica NEGAÇÃO (!)

```
1 boolean a = true;
2 !a equivalente a false
```


Programação estruturada com Java

Conversões

- Implícita - quando a variável que recebe a atribuição possui um tipo mais abrangente que o valor que será recebido

```
1 float a;  
2 int b = 5;  
3 a = b;
```

Programação estruturada com Java

Conversões

- Explícita (ou cast) - quando a variável que recebe a atribuição possui um tipo menos abrangente que o valor que será recebido

```
1 float a = 5.3 f;  
2 int b;  
3 b = (int) a;
```

Programação estruturada com Java

Conversões

- Convertendo String para número

```
1 String s = '8';  
2 int b;  
3 b = Integer.parseInt(s);
```

Programação estruturada com Java

Entrada e saída de dados

Entrada a partir da linha de comando

```
1 Scanner s = new Scanner(System.in);  
2 s.nextInt();  
3 s.nextLine();
```

Saída a partir da linha de comando

```
1 System.out.println(' 'Texto');
```

Programação estruturada com Java

Entrada e saída de dados

Exemplo

```
1 import java.util.Scanner;
2
3 public class EntradaSaida01 {
4     public static void main(String[] args) {
5         Scanner s = new Scanner(System.in);
6
7         System.out.println(" Digite seu nome");
8         String nome = s.nextLine();
9
10        System.out.println(" Digite sua idade");
11        int idade = s.nextInt();
12
13        System.out.println(nome + " tem " + idade + "
14                               anos.");
15    }
```

Programação estruturada com Java

Entrada e saída de dados

Entrada a partir de um painel

```
1 JOptionPane.showInputDialog(" texto" );
```

Saída a partir de um painel

```
1 JOptionPane.showMessageDialog( null , " texto" );
```

Programação estruturada com Java

Entrada e saída de dados

Exemplo

```
1 import javax.swing.JOptionPane;
2
3 public class EntradaSaida02 {
4     public static void main(String[] args) {
5         String nome =
6             JOptionPane.showInputDialog("Digite seu
7             nome");
8         String idadeStr =
9             JOptionPane.showInputDialog("Digite sua
10            idade");
11         int idade = Integer.parseInt(idadeStr);
12
13         JOptionPane.showMessageDialog(null, nome + " tem
14            " + idade + " anos.");
15     }
16 }
```

Programação estruturada com Java

Strings

- Strings são classes em Java, portanto possuem métodos já implementados
- O operador “+” serve para concatenação de strings
- Alguns métodos:
 - equals - verifica se duas strings possuem o mesmo conteúdo
 - toLowerCase - transforma o conteúdo da string para caixa baixa
 - toUpperCase - transforma o conteúdo da string para caixa alta
 - length() - retorna o tamanho da string

Programação estruturada com Java

Entrada e saída de dados

Exemplo

```
1 public class ExString {
2     public static void main(String[] args) {
3         String str      = "Seu nome é ";
4         String nome1     = "Alexandre";
5         String sobrenome = "Barbosa";
6         String nome2     = "Alexandre";
7         String nome3     = new String("Alexandre");
8
9         System.out.println(str + nome1 + sobrenome);
10        System.out.println(nome1.toLowerCase());
11        System.out.println(nome1.toUpperCase());
12        System.out.println(nome1.equals(sobrenome));
13        System.out.println(nome1 == sobrenome);
14        System.out.println(nome1.equals(nome2));
15        System.out.println(nome1 == nome2);
16        System.out.println(nome1 == nome3);
17    }
```

Programação estruturada com Java

Comentários em Java

Exemplo

```
1 public class Comentarios {
2     /**
3      * raiz – calcula raiz quadrada e retorna o valor
4      * @param i – o número para o qual se deseja
5      * calcular a raiz quadrada
6      * @return – um double correspondendo a raiz
7      * quadrada de i
8      */
9     public static double raiz(float i) {
10         // calcula raiz quadrada e retorna o valor
11         return Math.sqrt(i);
12     }
13
14     /**
15      * @param args – argumentos da linha de comando
16      */
17     public static void main(String[] args) {
```

Programação estruturada com Java

Parametros na linha de comando

Exemplo

```
1 public class LinhaComando {  
2  
3     public static void main(String[] args) {  
4         System.out.println(args[0]);  
5         System.out.println(args[1]);  
6     }  
7  
8 }
```

Programação estruturada com Java

Resumo

Resumo

- Java é uma linguagem multiplataforma e orientada a objetos, que foi criada com base na linguagem C++
- O projeto da linguagem tornou alguns conceitos transparentes ao programador (ex. ponteiros e alocação de memória)
- Todo programa Java deve ser criado em uma classe
- Cada classe deve ser definida em seu próprio arquivo
- Os tipos de dados primitivos existentes em Java são: byte, short, int, long, float, double e char
- Os tipos de dados representados por classes em Java são: Byte, Short, Integer, Long, Float, Double, Character e String

Programação estruturada com Java

Resumo

Resumo

- Operadores relacionais Java são os mesmos adotados em C: <, >, <=, >=, !=, == e =
- Operadores matemáticos Java são os mesmos adotados em C: +, -, /, % e *
- Operadores lógicos Java são representados por: && (E lógico), || (OU lógico) e !(Negação lógica)
- Entrada e saída em Java pode ser realizada pelo console (System / in / out / Scanner)
- Entrada e saída em Java pode ser realizada através da classe JOptionPane

Programação estruturada com Java

Exercícios

Exercício

- ❶ Crie programas em Java para:
 - ❶ receber dois valores fornecidos pelo usuário e apresentar o resultado das quatro operações
 - ❷ calcular a média de duas notas fornecidas pelo usuário
 - ❸ apresentar um menu de opções para o usuário, onde cada opção deve ser exibida em uma linha diferente
 - ❹ ler dois valores fornecidos pelo usuário, armazenar nas variáveis A e B, e realizar a troca do conteúdo destas
 - ❺ ler um texto e um caractere fornecidos pelo usuário e indicar se o caractere está presente ou não no texto
 - ❻ ler um texto e um caractere fornecidos pelo usuário e indicar qual a primeira posição que o caractere é exibido
 - ❼ ler um texto e um caractere fornecidos pelo usuário e substituir as ocorrências do caractere no texto

Para observar funções úteis, visite: <http://docs.oracle.com/javase/6/docs/api/java/lang/String.html>

Programação estruturada com Java

Exercícios

Exercício

- 1 Revise a implementação dos algoritmos implementados nas disciplinas P1 e LP1

Perguntas?

Alexandre de Andrade Barbosa
alexandre.barbosa@arapiraca.ufal.br